

BACKGROUND OF THE INVENTION

1. Field of the Invention

5 This invention relates to computer systems and, more particularly, to network interconnection hardware and software.

2. Description of the Related Art

10 The use of modern computer systems is placing an increased demand on computer system network bandwidth. Higher performance system servers, mass storage devices and input/output devices with higher bandwidth and lower latency have been outpacing the existing interconnection technologies. Thus a system interconnection solution was needed that would overcome some of the bandwidth and latency problems associated
15 with existing interconnection technologies.

 One such interconnection solution is the Infiniband™ Switched Fabric. The Infiniband architecture is a point-to-point interconnection fabric where nodes are interconnected via switching devices. In particular, the Infiniband architecture describes
20 a system area network in which independent processing nodes may be interconnected with I/O devices. The Infiniband architecture is described in detail in the Infiniband™ Architecture Specification available from the Infiniband™ trade association.

25

The Infiniband specification defines an interface between a processing node operating system and the node's hardware interface to the fabric. The hardware interface on a processing node is referred to as a hardware channel adapter. One fundamental idea behind Infiniband is the ability of a client process to place instructions in a queue, for the hardware to execute. The queue is referred to as a work queue. Each work queue is created in a pair called a queue pair. Each queue pair has a send queue and a receive queue. The queue pair creates a virtual communication port for a client process to communicate with other processes and end nodes. A queue pair is an abstract construct of memory locations and may have a predetermined number of entries that hold a predetermined number of work queue elements. An instruction is stored in the queue pair in the form of a work queue element. The hardware channel adapter services work queue elements in the order they are received into the queue pair.

Although the Infiniband specification describes a channel interface between the operating system and the fabric, the Infiniband specification does not specify how the hardware channel adapter or the software driving the hardware channel adapter must be implemented. Therefore, a solution is needed to bridge a host processing node to the Infiniband fabric.

20

SUMMARY OF THE INVENTION

Various embodiments of a method and apparatus for manipulating work queue elements are disclosed. In one embodiment, a hardware channel adapter may service work queue elements that are stored in a queue pair. The hardware channel adapter may include hardware registers that may track work queue elements that are currently being serviced and which work queue element will be serviced next. A software driver may cause the work queue elements to be stored in the queue pair. The software driver notifies the hardware channel adapter when there are new work queue elements to service. Additionally, each work queue element may include an indication of whether the work queue element has completed. The software driver may cause a new work queue element to be stored in the location previously occupied by the completed work queue element upon detecting the completion indication. Thus, the combination of the hardware channel adapter and the software driver may allow for work queue elements to be serviced in a first come first served manner even if the work queue elements become non-contiguous in memory due to the availability of free locations resulting from out of order completions and subsequent use of those free locations by the software driver as it adds new work queue elements to the queue.

Broadly speaking, in one embodiment, an apparatus including a software driver and a hardware adapter is contemplated. The software driver is configured to cause a plurality of work queue elements to be stored in a queue pair including a plurality of storage locations. Each of the plurality of storage locations includes an indicator indicating whether a corresponding work queue element has been completed. The hardware adapter is configured to select one of the plurality of storage locations and to service a corresponding one of the plurality of work queue elements, and in response to completion of a task associated with the corresponding work queue element, to cause the

indicator to indicate that the corresponding work queue element has been completed. Additionally, the software driver is configured to cause a new work queue element to be stored in the selected storage location in response to detecting that the indicator indicates that the corresponding work queue element has been completed.

5

In one particular implementation, the hardware adapter includes a first register for storing a virtual address of the selected storage location and the corresponding work queue element. The hardware adapter further includes a second register for indicating a number of pending work queue elements remaining to be serviced. The software driver is further configured to notify the hardware adapter when the new work queue element is stored by causing the virtual address of the new work queue element to be written to the first register of the hardware adapter. The hardware adapter is further configured to increment the second register in response to receiving the notification from the software driver.

15

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of one embodiment of a computer network.

20

FIG. 2 is a functional block diagram of one embodiment of a host platform.

FIG. 3A is a diagram of one embodiment of a hardware channel adapter.

25

FIG. 3B is a diagram of one embodiment of a queue pair data structure.

FIG. 4 is a flow diagram of the operation of one embodiment of a software driver.

FIG. 5 is a flow diagram of the operation of one embodiment of a hardware channel adapter.

While the invention is susceptible to various modifications and alternative forms,
5 specific embodiments thereof are shown by way of example in the drawings and will
herein be described in detail. It should be understood, however, that the drawings and
detailed description thereto are not intended to limit the invention to the particular form
disclosed, but on the contrary, the intention is to cover all modifications, equivalents and
alternatives falling within the spirit and scope of the present invention as defined by the
10 appended claims.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Turning now to FIG. 1, a block diagram of one embodiment of a computer network is shown. The computer network includes a first host platform 10A, a second host platform 10B and a mass storage device 20 each interconnected by a switch 30A. Additionally, an input/output chassis 40 is connected to the network via a second switch 30B. Each of the devices connected by one of switch 30A or 30B may be referred to as an end node or simply a node. The network may be interconnected by a switched interconnection fabric 50.

In one embodiment, the switched interconnection fabric 50 is an Infiniband™ switched interconnection fabric. Host platform 10A and 10B may each be a computing node such as a personal computer or a workstation, for example. Host platform 10A and 10B may each run one or more client processes either independently or in a shared processing environment and may communicate with each other, with I/O chassis 40 and with mass storage 20 via switch 30A and switch 30B. I/O chassis 40 is an exemplary function that may include one or more types of input/output device controllers. Mass storage 20 may be one of many types of storage devices and may be configured as a stand-alone device or in an array such as a hard disk array, a random access memory array or a tape drive array. Switch 30A and 30B may be configured to receive and forward packets of information to the next destination. It is noted that in other embodiments, other configurations and numbers of devices are contemplated.

As will be described in more detail below, host platform 10 may include a hardware channel adapter, which physically connects host platform 10 to the Infiniband fabric and a software driver that provides a software interface between the operating system of host platform 10 and the hardware channel adapter. A host process running on

host platform 10 may generate instructions that the hardware channel adapter ultimately executes.

Referring to FIG. 2, a functional block diagram of one embodiment of a host platform 10 of FIG. 1 is shown. Host platform 10 may include one or more processors (not shown) that execute host processes such as process 100A and a process 100B. Process 100A and 100B are operatively coupled to a memory 200 through a channel adapter software driver 150. A hardware channel adapter 175 is coupled to software driver 150 and to memory 200.

10

In one embodiment, process 100A and 100B (hereafter 100) may generate instructions to be executed by hardware channel adapter 175. For example, process 100 may generate instructions to move data from one location in memory 200 to another location in memory 200. Additionally, process 100 may generate instructions to move data from one location in memory 200 to an additional memory located on another node.

15

In one particular implementation, process 100 makes a request to communicate to a particular device or process and software driver 150 may respond by allocating space in memory 200 to create a queue pair corresponding to that device or process. Each time process 100 makes a request for the hardware adapter to perform a particular function, software driver 150 places the request in the corresponding queue pair as a work queue element. Software driver 150 then notifies hardware adapter 175 that there is a work request in the queue. Software driver 150 may cause work queue elements to be stored in the queue pair in a sequential order until such time as the queue begins to fill up.

20
25

Hardware adapter 175 may be configured to fetch a work queue element from the queue pair in memory 200 and to execute the instructions contained therein. Hardware adapter 175 may service each work queue element in the order it is placed into the queue.

However, the work queue element may not complete in the same order that it was executed. A completion queue (not shown) may hold a list of work queue elements that have completed and an indication may be presented to both hardware adapter 175 and software driver 150 indicating the availability of such free locations. As a work queue element completes, the location in the queue pair in which it was stored, may become available. Software driver 150 may place new work queue elements in these completed out of order queue locations. Thus, the combination of the hardware channel adapter 175 and software driver 150 may allow for work queue elements to be serviced in a first come first served manner even if the work queue elements become non-contiguous in memory due to the availability of free locations resulting from out of order completions and subsequent use of those free locations by software driver 150 as it adds new work queue elements to the queue. It is noted that software driver 150 may create the queue pairs and any associated software registers using a linked list data structure or any other form of data structure that may achieve similar operational characteristics as the implementations described herein.

Turning now to FIG. 3A, a diagram of one embodiment of the hardware channel adapter 175 of FIG. 2 is shown. Hardware channel adapter 175 of FIG. 3A includes a plurality of hardware registers for managing the work queue elements stored in various queue pairs in memory. In the illustrated embodiment, hardware channel adapter 175 includes a Next_VA_Reg, an HW_Doorbell_Reg and an In_Service_Bit for each corresponding queue pair that has been created by the software driver 150 of FIG. 2. For example, QP1_Next_VA_Reg corresponds to queue pair 1 and QP2_Next_VA_Reg corresponds to queue pair 2.

25

The Next_VA_Reg register is a hardware register used to store the virtual address of the next work queue element to be serviced by hardware channel adapter 175. The HW_Doorbell_Reg is a hardware register used to store a count of the number of work

queue elements remaining to be serviced by hardware channel adapter 175. The In_Service_Bit is used to keep track of whether the work queue element at the address stored in the Next_VA_Reg register has been serviced. The In_Service_Bit, while set, indicates that the work queue element is still active in the fabric and has not yet
5 completed. Thus preventing a work queue element from being serviced more than once
Operation of hardware channel adapter 175 of FIG. 3A will be described in greater detail below.

Referring to FIG 3B, a diagram of one embodiment of a queue pair 210 data
10 structure is shown. Queue pair 210 is implemented in a group of memory locations and created by the software driver 150 of FIG. 2 to store information corresponding to work queue elements. In FIG. 3B, QPx 210 is representative of any number of queue pairs that may be created by software driver 150. In this exemplary diagram, QPx 210 includes four entries available for storing work queue elements and they are labeled WQE_VA0-3.
15 Each entry in QPx 210 includes five fields: Virtual Address, Work Queue Element Information, Next Virtual Address, Next Work Queue Element Posted, and Busy. Additionally, a software register, Driver Last Work Queue Element Accessed 215 corresponding to QPx 210 is shown. In the illustrated embodiment, QPx 210 has four entries. It is contemplated that other embodiments may use more or less entries and there
20 may be a Driver Last Work Queue Element Accessed register for each queue pair created by software driver 150. Operation of software driver 150 and the use of QPx 210 will be described in greater detail below.

Turning to FIG. 4, a flow diagram of the operation of one embodiment of software
25 driver 150 of FIG. 2 is shown. Referring collectively to FIG. 2, FIG. 3A and FIG. 3B, the operation of software driver 150 of FIG. 2 is described. The flow diagram of FIG. 4 begins in step 400 where software driver 150 is in an idle state. If a new work request is made to software driver 150 or a work queue element completes, operation proceeds to

step 410 where software driver 150 checks the request to see if the request is a new work request. If the request is not a new work request, operation proceeds to step 450 of FIG. 4 where software driver 150 checks a completion queue for the completed work queue element. Software driver 150 then clears the Busy bit in FIG. 3A for that corresponding work queue element. Operation then proceeds back to step 400 of FIG. 4. Referring back to step 410, if the request is a new work request, then operation proceeds to step 420. In step 420, software driver 150 checks for an available location in a queue pair such as QPx 210 of FIG. 3A by checking the Busy bit of each location in QPx 210. If there are no available locations, operation proceeds back to step 400 of FIG. 4 where software driver 150 may continue to try to deposit the new work queue element into QPx 210 at predetermined intervals. In step 420, if there are locations available such as location WQE_VA0, for example, then operation proceeds to step 430. In step 430, software driver 150 writes the work queue element into location WQE_VA0. Software driver 150 then sets the Busy bit, which indicates that this location is no longer available. Software driver 150 then writes the virtual address of the last location (e.g. WQE_VA0) in software register Last Work Queue Element Accessed 215 of FIG. 3A. Software driver 150 then goes back to location WQE_VA0 and sets the Next WQE Posted Bit and writes the virtual address of the next available location (e.g. WQE_VA1) into the Next Virtual Address field of QPx 210. Operation then proceeds to step 440 of FIG. 4. Software driver 150 rings the hardware doorbell by writing to a corresponding QPx HW_Doorbell_Reg of FIG. 3B, thus notifying hardware channel adapter 175 of a new work request. Operation then proceeds back to step 400.

Referring now to FIG. 5, a flow diagram of the operation of one embodiment of hardware channel adapter 175 of FIG. 3A is shown. The operation of hardware channel adapter 175 begins in step 500 of FIG. 5. In step 500, hardware channel adapter 175 checks QPx HW_Doorbell_Reg of FIG. 3B see if the count is not equal to zero. If the count is zero, then operation stays in step 500. However, if the count is not zero, then

operation proceeds to step 510 where hardware channel adapter 175 checks to see if there is a new doorbell by checking to see if software driver 150 has written to QPx_Next_VA_Reg of FIG. 3A. If there is a new doorbell, operation proceeds to step 520 of FIG. 5, where hardware channel adapter 175 increments the HW_Doorbell_Reg.

- 5 Operation then proceeds to step 530, where hardware channel adapter 175 checks to see if it is the first doorbell ring for that queue pair. The first doorbell ring notifies the hardware of the starting point of that particular queue pair in memory. Since queue pairs and work queue elements may be a predetermined size, the hardware may now track where each successive work queue element is located. If it is not the first doorbell ring,
- 10 operation proceeds to step 550. Going back to step 530, if it is the first doorbell ring, operation proceeds to step 540, where hardware channel adapter 175 copies data written to QPx_Next_VA_Reg of FIG. 3B into QPx_Next_VA_Reg and operation proceeds to step 550. Going back to step 510, if there is no new doorbell and the HW_Doorbell_Reg of FIG. 3B is not zero, operation proceeds to step 550 of FIG. 5. In step 550, hardware
- 15 channel adapter 175 checks if necessary resources are available to service a work queue element. If the resources are not available, operation proceeds back to step 500. If the resources are available, operation proceeds to step 560 where hardware channel adapter 175 fetches the virtual address in QPx_Next_VA_Reg of FIG. 3B and reads the work queue element stored there. Operation then proceeds to step 570. Hardware channel
- 20 adapter 175 then checks the In_Service_Bit of FIG. 3A. If the In_Service_Bit was not set, operation proceeds to step 580. In step 580, hardware channel adapter 175 places the work queue element into service, sets the In_Service_Bit of FIG. 3A and decrements the HW_Doorbell_Reg. Operation then proceeds back to step 500 of FIG. 5. Referring back to step 570, if the In_Service_Bit of FIG. 3A is set, then operation proceeds to step 590 of
- 25 FIG. 5. In step 590, hardware channel adapter 175 clears the Next_WQE_Posted bit of FIG. 3B and loads the Next Virtual Address into QPx_Next_VA_Reg of FIG. 3A. Operation then proceeds back to step 500.

1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2